

### IN THE CLAIMS

Please amend the claims as follows:

- 1) (Currently amended) A method for comparing two program source code files to help an expert determine whether one file contains source code that has been copied from the other file or whether both files contain code that has been copied from a third file, the method comprising
  - a) eliminating programming comments from the first source code file;
  - b) eliminating programming comments from the second source code file;
  - c) substituting a single space character for sequences of whitespace characters in each remaining line of functional programming code in said first file;
  - d) substituting a single space character for sequences of whitespace characters in each remaining line of functional programming code in said second file;
  - e) eliminating all lines of functional programming code in said first file that consist entirely of programming keywords;
  - f) eliminating all lines of functional programming code in said second file that consist entirely of programming keywords;
  - g) putting each remaining line of functional programming code of the first file into a first array of text strings;
  - h) putting each remaining line of functional programming code of the second file into a second array of text strings; ~~and~~
  - i) finding all matches between text strings in said first array with text strings in said second array; and
  - j) displaying a report showing said matches.

- 2) (Previously presented) The method of claim 1) where finding all matches ignores the type case of the text.
- 3) (Currently amended) ~~A method for comparing two program source code files to help an expert determine whether one file contains source code that has been copied from the other file or whether both files contain code that has been copied from a third file~~ The method of claim 1), the method further comprising
  - a) eliminating functional programming lines from ~~the~~ said first source code file, leaving comment lines;
  - b) eliminating functional programming lines from ~~the~~ said second source code file, leaving comment lines;
  - c) substituting a single space character for sequences of whitespace characters in each remaining comment line in said first file;
  - d) substituting a single space character for sequences of whitespace characters in each remaining comment line in said second file;
  - e) putting each remaining comment line of the first file into an first array of text strings;
  - f) putting each remaining comment line of the second file into a second array of text strings; ~~and~~
  - g) finding all matches between text strings in said first array with text strings in said second array-; and  
displaying a report showing said comment line matches.
- 4) (Currently amended) The method of claim 3) where finding all comment line matches ignores the type case of the text.
- 5) (Currently amended) A method for comparing two program source code files to help an expert determine whether one file contains source code that has been copied from the other file or whether both files

contain code that has been copied from a third file, the method comprising

a) extracting all words between whitespace from each line of functional programming code in the first source code file to a first array of text strings;

b) eliminating programming language keywords from said first array of text strings;

eliminating all words from said first array of text strings that are less than a minimum length of characters;

c) extracting all words between whitespace from each line of functional programming code in the second source code file to a second array of text strings;

d) eliminating programming language keywords from said second array of text strings;

eliminating all words from said second array of text strings that are less than a minimum length of characters;

e) finding all matches between text strings in said first array with text strings in said second array; and

displaying a report showing said matches.

6) (Previously presented) The method of claim 5) where finding all matches ignores the type case of the text.

7) (Currently amended) A method for comparing two program source code files to help an expert determine whether one file contains source code that has been copied from the other file or whether both files contain code that has been copied from a third file, the method comprising

- a) extracting all words between whitespace from each line of functional programming code in the first source code file to an first array of text strings;
  - b) eliminating programming language keywords from said array of text strings;
  - e) extracting all words between whitespace from each line of functional programming code in the second source code file to a second array of text strings;
  - d) eliminating programming language keywords from said second array of text strings;
  - e) finding all partial matches between text strings in said first array with text strings in said second array, where a partial match is where one string can be found in its entirety in ~~as~~ a second string but the strings are not identical; and  
displaying a report showing said partial matches.
- 8) (Previously presented) The method of claim 7) where finding all partial matches ignores the type case of the text.
- 9) (Currently amended) A method for comparing two program source code files to help an expert determine whether one file contains source code that has been copied from the other file or whether both files contain code that has been copied from a third file, the method comprising
- a) eliminating programming comments from the first source code file;
  - b) eliminating programming comments from the second source code file;
  - e) substituting a single space character for sequences of whitespace characters in each remaining line of functional programming code in said first file;

- e) substituting a single space character for sequences of whitespace characters in each remaining line of functional programming code in said second file;
  - e) putting each remaining line of functional programming code of the first file into a first array of text strings;
  - f) putting each remaining line of functional programming code of the second file into a second array of text strings; ~~and~~
  - g) finding sequences where the first word of each line in said first array matches the first word of each line in said second array; and  
displaying a report showing said sequences.
- 10) (Previously presented) The method of claim 9) where finding sequences where the first word of each line in said first array matches the first word of each line in said second array ignores the type case of the text.
- 11) (Currently amended) A method for comparing two program source code files, comprising:
- a) extracting from each program source code file a first set of code elements and a second set of code elements;
  - b) computing a first metric derived from comparing the first set of code elements for the first program source code file to the first set of code elements for the second program source code file;
  - e) computing a second metric derived from comparing the second set of code elements for the first program source code file to the second set of code elements for the second program source code file;
  - e) combining the first metric and the second metric to derive a combined metric, wherein the first and second sets of code elements are selected from the group consisting of complete

words, selected partial words, selected source lines, selected comment lines and selected code sequences-; and

displaying a report showing said combined metric.

- 12) (Currently amended) An apparatus for comparing two program source code files to help an expert determine whether one file contains source code that has been copied from the other file or whether both files contain code that has been copied from a third file, the apparatus comprising

A computer;

A source code matching program on said computer, wherein said source code matching program comprises:

a) means for eliminating programming comments from the first source code file;

b) means for eliminating programming comments from the second source code file;

c) means for substituting a single space character for sequences of whitespace characters in each remaining line of functional programming code in said first file;

d) means for substituting a single space character for sequences of whitespace characters in each remaining line of functional programming code in said second file;

means for eliminating all lines of functional programming code in said first file that consist entirely of programming keywords;

means for eliminating all lines of functional programming code in said second file that consist entirely of programming keywords;

e) means for pPutting each remaining line of functional programming code of the first file into a first array of text strings;

~~f)~~ means for putting each remaining line of functional programming code of the second file into a second array of text strings; and

~~g)~~ means for finding all matches between text strings in said first array with text strings in said second array~~;~~ and

means for displaying a report showing said matches.

13) (Currently amended) The apparatus of claim ~~11)~~ 12) where means for finding all matches ignores the type case of the text.

14) (Currently amended) ~~An apparatus for comparing two program source code files to help an expert determine whether one file contains source code that has been copied from the other file or whether both files contain code that has been copied from a third file~~ The apparatus of claim 12), the apparatus further comprising

A computer;

A source code matching program on said computer, wherein said source code matching program comprises:

~~a)~~ means for eliminating functional programming lines from ~~the~~ said first source code file, leaving comment lines;

~~b)~~ means for eliminating functional programming lines from ~~the~~ said second source code file, leaving comment lines;

~~c)~~ means for substituting a single space character for sequences of whitespace characters in each remaining comment line in said first file;

~~d)~~ means for substituting a single space character for sequences of whitespace characters in each remaining comment line in said second file;

~~e)~~ means for putting each remaining comment line of the first file into ~~a~~ first array of text strings;

~~f~~ means for putting each remaining comment line of the second file into a second array of text strings; ~~and~~

~~g~~ means for finding all matches between text strings in said first array with text strings in said second array; and

means for displaying a report showing said comment line matches.

15) (Currently amended) The apparatus of claim 14) where means for finding all comment line matches ignores the type case of the text.

16) (Currently amended) An apparatus for comparing two program source code files to help an expert determine whether one file contains source code that has been copied from the other file or whether both files contain code that has been copied from a third file, the apparatus comprising

A computer;

A source code matching program on said computer, wherein said source code matching program comprises:

~~a~~ means for extracting all words between whitespace from each line of functional programming code in the first source code file to ~~a~~ first array of text strings;

~~b~~ means for eliminating programming language keywords from said first array of text strings;

means for eliminating all words from said first array of text strings that are less than a minimum length of characters;

~~e~~ means for extracting all words between whitespace from each line of functional programming code in the second source code file to a second array of text strings;

means for eliminating all words from said second array of text strings that are less than a minimum length of characters;



~~d)~~ means for eliminating programming language keywords from said second array of text strings;

~~e)~~ means for finding all matches between text strings in said first array with text strings in said second array; and

means for displaying a report showing said matches.

17) (Previously presented) The apparatus of claim 16) where means for finding all matches ignores the type case of the text.

18) (Currently amended) An apparatus for comparing two program source code files to help an expert determine whether one file contains source code that has been copied from the other file or whether both files contain code that has been copied from a third file, the apparatus comprising

A computer;

A source code matching program on said computer, wherein said source code matching program comprises:

~~a)~~ means for extracting all words between whitespace from each line of functional programming code in the first source code file to a first array of text strings;

~~b)~~ means for eliminating programming language keywords from said first array of text strings;

~~e)~~ means for extracting all words between whitespace from each line of functional programming code in the second source code file to a second array of text strings;

~~d)~~ means for eliminating programming language keywords from said second array of text strings;

~~e)~~ means for finding all partial matches between text strings in said first array with text strings in said second array, where a

partial match is where one string can be found in its entirety in  
as a second string but the strings are not identical~~+~~; and  
means for displaying a report showing said partial matches.

- 19) (Previously presented) The apparatus of claim 18) where means for  
finding all partial matches ignores the type case of the text.
- 20) (Currently amended) An apparatus for comparing two program source  
code files to help an expert determine whether one file contains  
source code that has been copied from the other file or whether both  
files contain code that has been copied from a third file, the  
apparatus comprising

A computer;

A source code matching program on said computer, wherein said source  
code matching program comprises:

a~~+~~ means for eliminating programming comments from the first source  
code file;

b~~+~~ means for eliminating programming comments from the second source  
code file;

e~~+~~ means for substituting a single space character for sequences of  
whitespace characters in each remaining line of functional  
programming code in said first file;

d~~+~~ means for substituting a single space character for sequences of  
whitespace characters in each remaining line of functional  
programming code in said second file;

e~~+~~ means for putting each remaining line of functional programming  
code of the first file into an first array of text strings;

f~~+~~ means for putting each remaining line of functional programming  
code of the second file into a second array of text strings; ~~and~~

g) means for finding sequences where the first word of each line in said first array matches the first word of each line in said second array; and

means for displaying a report showing said sequences.

21) (Previously presented) The apparatus of claim 20) where means for finding sequences where the first word of each line in said first array matches the first word of each line in said second array ignores the type case of the text.

22) (Currently amended) An apparatus for comparing two program source code files, comprising:

A computer;

A source code matching program on said computer, wherein said source code matching program comprises:

a) means for extracting from each program source code file a first set of code elements and a second set of code elements;

b) means for computing a first metric derived from comparing the first set of code elements for the first program source code file to the first set of code elements for the second program source code file;

c) means for computing a second metric derived from comparing the second set of code elements for the first program source code file to the second set of code elements for the second program source code file;

d) means for combining the first metric and the second metric to derive a combined metric, wherein the first and second sets of code elements are selected from the group consisting of complete words, selected partial words, selected source lines, selected comment lines and selected code sequences; and

means for displaying a report showing said metrics.